Atty Docket No. 29250-000601
Lucent Case Name: Hu 6-24

METHOD AND APPARATUS FOR TRANSMITTING AND RECEIVING DATA PACKETS TO AVOID STALL DURING RE-SEQUENCING OF DATA PACKETS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to next generation wireless communication; and more particularly, to eliminating stall during the re-sequencing of received data packets.

2. Description of Related Art

In telecommunications systems, particularly next generation wireless packet data systems, a base station transmits data packets to user equipments. However, the user equipment does not necessarily receive the data packets in the same order as they were transmitted. This out-of-sequence problem is compounded in systems such as High Speed Downlink Packet Access (HSDPA) in which an in-band priority class indicator (PCI) has been proposed to differentiate between the priority of logical channels' multiplexed in the same transport channel. Namely, even though data packets of a given priority class are transmitted in order, data packets of the given priority class will be received out of order on the receive side. In

addition, data packets of other priority classes will be received interspersed with the receipt of data packets for the given priority class.

It has also been proposed with HSDPA that an in-band transmission sequence number (TSN) be provided with data packets for the re-sequencing functionality on the received side. And, U.S. Patent Application No. unknown entitled "A MULTI-PRIORITY RE-SEQUENCING METHOD AND APPARATUS", which is hereby incorporated by reference in its entirety, by the inventors of the subject application, discloses several methods of re-sequencing received data packets.

It is also the case that data packets are not received in certain types of systems. For example, when the receiver at a destination device (e.g., the user equipment) fails to decode a data packet, and sends the source device (e.g., the base station) of the data packet a non-acknowledge response indicating failure to properly receive the data packet, the source device is expected to retransmit the data packet. However, a transmission error (hereinafter "acknowledgement error") can occur with the non-acknowledge response such that the source device interprets the response as an acknowledge response – indicating receipt of the data packet by the destination device. As a result, the source device does not retransmit the data packet, and a gap will exist in the sequence of data packets at the destination device. A gap in the sequence of data packets at the destination device is also created if a data packet of a lower priority class has its transmission aborted in favor of a data packet of a higher priority class.

U.S. Patent Application No. unknown entitled "A MULTI-PRIORITY RE-SEQUENCING METHOD AND APPARATUS" by the inventors of the subject application discloses using a timer or counter to flush gaps in the sequence of data packets. The delay in outputting data packets due to a missing packet is called a stall. The stall period is determined by the length of the timer, and is typically set for a worst-case transmission scenario. For

example, it may take several re-transmission attempts for proper receipt of a data packet. The length of the count down timer can be set to permit a maximum number of these retransmission attempts. However, the acknowledgement error and data packet abortion scenarios described above occur very quickly – in much less time than the length of the count down timer. Consequently, a lengthy stall occurs.

SUMMARY OF THE INVENTION

The present invention provides a method and apparatus for avoiding stalls in the re-sequencing of data packets.

When the source device receives an acknowledgement that the destination device has received a data packet, the source device inserts an acknowledge sequence number in a data packet for transmission. The acknowledge sequence number is the same as the transmission sequence number of the acknowledged data packet. On the receive side, the destination device will treat a missing data packet having the same transmission sequence number as the acknowledge sequence number as having been output to the next, upper layer of processing; namely, confirmation that the missing data packet is lost. And, because the missing data packet is flushed, stall is avoided.

Furthermore, when the transmission of a data packet is aborted, the source device inserts an acknowledge sequence number in a data packet for transmission. The acknowledge sequence number is the same as the transmission sequence number of the aborted data packet. Consequently, on the receive side, the destination device will treat the missing, aborted data packet having the same transmission sequence number as the acknowledge sequence number as having been output to the next, upper layer of processing. Because the missing, aborted data packet is flushed, stall is avoided.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will become more fully understood from the detailed description given herein below and the accompanying drawings which are given by way of illustration only, wherein like reference numerals designate corresponding parts in the various drawings, and wherein:

Fig. 1 illustrates a logical block diagram of a transmit and receive system employing stall avoidance methodologies according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERED EMBODIMENTS

Fig. 1 illustrates a logical block diagram of a transmit and receive system employing stall avoidance methodologies embodiment of the present invention. As shown, a source device (e.g., a base station) 10 transmits data packets over a medium 12 to a destination device (e.g., user equipment) 14. Hereinafter, the source device 10 will be referred to as base station 10 and the destination device 14 will be referred to as the user equipment 14, but it will be understood that the user equipment could be the source device and the base station the destination device.

The base station 10 includes a parser 2 that receives data packets for transmission from other base station circuitry, not shown for the sake of brevity and clarity. The data packets represent one or more logical channels, and have been assigned a priority class. Specifically, each data packet includes a priority class indicator (PCI) indicating the priority class (1 through N) of the data packet. In one embodiment, the PCI is disposed in a header portion of the data packet.

The parser 2 parses the data packets into one of N buffers 4 based on the PCI of the data packet. A buffer 4 is provided for, and associated with, a different one of the N priority classes; hence, N buffers 4 are provided. For

example, the parser 2 outputs data packets having a PCI of 1 to the buffer 4 associated with priority class 1, outputs data packets having a PCI of 2 to the buffer 4 having a priority class of 2, etc.

A scheduler 9 receives the data packets from the buffers 4, and schedules the data packets for transmission by a transmitter 8. Specifically, a scheduling unit 3 receives the data packets, and orders the data packets for transmission based on their priority class. Data packets having a higher priority class will be scheduled for transmission prior to data packets of a lower class, and data packets having a higher priority class will pre-empt the transmission of data packets having a lower priority class. If a data packet having a higher priority class than the data packet currently undergoing transmission/re-transmission is received by the scheduling unit 3, the scheduling unit 3 instructs the transmitter 8 to abort transmission of the lower priority class data packet. The scheduling unit 3 also notifies an ASN generator 7 (discussed in detail below) of the aborted data packet.

A transmission sequence number (TSN) generator 6 receives the data packets for transmission from the scheduling unit 3 and assigns each data packet a transmission sequence number. The TSN generator 6 keeps track of a transmission sequence for each priority class; accordingly, there are N transmission sequences. The TSN generator 6 assigns the next TSN in the transmission sequence for the priority class to which a received data packet belongs. Specifically, the TSN is disposed in a predetermined position in the data packet. For example, in one embodiment, the TSN forms part of the header for the data packet.

After assigning a TSN to a data packet, the TSN generator 6 increments the TSN, and this new TSN will be assigned to the next, received data packet of that priority class. In a preferred embodiment, the TSN generator 6 begin by assigning a TSN of, for example, 0, and increments the TSN by 1.

Because a different transmission sequence is associated with each priority class, the data packets of a particular priority class have TSNs that are independent of the TSNs assigned to data packets of other priority classes. As a result, two data packets of different priority classes can have the same TSN.

An acknowledge sequence generator 7 generates an acknowledge sequence number (ASN) based on information from a receiving system 13', and inserts the generated ASN into a predetermined position in the data packet. For example, in one embodiment, the ASN forms part of the header for the data packet. The receiving system 13' has the same structure as the receiving system 13 in the user equipment 14, and will, therefore, not be described in detail as the receiving system 13 is described in detail below. Briefly, however, the receiving system 13' notifies the ASN generator 7 of the TSN and PCI for data packets that have been acknowledged by the user equipment 14 as properly received. The receiving system 13' also notifies the transmitter 8 of the TSN and PCI of data packets that have been properly and improperly received by the user equipment 14 along with and indication of whether receipt was proper or improper.

The ASN generator 7 generates an ASN equal to the TSN of the acknowledged data packet, and stores the generated ASN by the priority class of the acknowledged data packet. The ASN generator 7 also generates an ASN equal to the TSN of the data packet aborted by the scheduling unit 3. Specifically, the scheduling unit 3 identifies the data packet to be aborted by its TSN and PCI, the ASN generator 7 generates the ASN equal to the TSN of the aborted data packet, and stores the generated ASN with the other ASNs having the same priority class as the aborted data packet. The ASN generator 7 receives a data packet for transmission from the TSN generator 6, and identifies its priority class from its PCI. The ASN generator 7 then looks up the stored ASNs having that priority class, and inserts, for

example, in one embodiment, the latest or most current stored ASN in the data packet. In this manner, the ASNs are inserted into the data packets on a last-in, first-out basis. In one embodiment, for example, the inserted ASN forms part of the header for the data packet.

A transmitter 8 receives the data packets from the scheduler 9, and multiplexes the data packets into a transport channel for transmission via a duplexer 11 and one or more antennas At over the medium 12. For data packets that the receiving system 13' identifies as not being properly received by the user equipment 14, the transmitter 8 retransmits those data packets. Additionally, the transmitter 8 expects to receive confirmation that a data packet has been received a predetermined period of time after transmission. If no such confirmation is received, then the transmitter 8 retransmits the data packet. And, after a data packet has been retransmitted a predetermined number of times, the transmitter 8 will not attempt further retransmissions.

At the user equipment 14, the data packets are received by a receiving system 13 via one or more receive antennas Ar and a duplexer 15. A receiver 16 decodes the data packets, and outputs the data packets to a single buffer 23, which stores the data packets. In a preferred embodiment, the buffer 23 is a random access memory. When the receiver 16 sends the data packets to the buffer 23, the receiver 16 instructs the transmission system 11' to output an acknowledge response for the received data packet. The acknowledge response identifies the received data packet by its PCI and TSN and indicates that the data packet has been received. However, if the receiver 16 is unable to receive the data packet (e.g., can not decode the data packet), the receiver 16 instructs the transmission system 11' to send a non-acknowledge response for the data packet. The non-acknowledge response identifies the received data packet by its PCI and TSN and indicates that the data packet has not been received. The transmission

system 11' has the same structure as the transmission system 11 in the base station 10 described in detail above. For the purposes of explanation only, it has been assumed that the same technology is implemented for the reverse link (user equipment 14 to base station 10) and the forward link (base station 10 to user equipment 14).

Control logic 25 in the user equipment 14 causes the buffer 23 to output data packets to the next, upper layer of processing based on the PCI and TSN of the data packets. Specifically, the control logic 25 includes a counter 27 associated with each priority class. Accordingly, there are N counters 27. After the buffer 23 has been loaded with a predetermined number of data packets, the control logic 25 examines the TSNs of the data packets for priority class 1 and initializes the counter 27 associated with that priority class to a value equal to the lowest TSN of the data packets having a PCI of 1 in the buffer 23. This process is then repeated for each priority class.

Next, the control logic 25 instructs the buffer 23 to output a data packet having a PCI of 1 and the same TSN as the value of the counter 27 for priority class 1. After the buffer 23 outputs such a data packet, the counter 27 is incremented by the same amount that the TSN generator 6 increments the TSN. The comparison, output and increment process is then repeated for other data packets having a priority class of 1.

Furthermore, the same comparison, output and increment process is performed for data packets in the other priority classes using their associated counters 27. Additionally, when a counter 27 is initialized or after the counter 27 is incremented, the control logic 25 starts or resets a count down timer (e.g., 10 seconds) 19 associated with the counter 27. Alternatively, the count down timer 19 is started when a data packet having a TSN greater than the count value of the counter 27 is received. If the count down timer 19 expires before the buffer 23 receives a data packet having a

TSN equal to the count value of the counter 27 for that priority class, the counter 27 is incremented. In this manner, even though data packets are received out of order by the user equipment 14, the data packets are sent to the next, upper layer of system processing in sequence.

The control logic 25 further monitors any ASNs in the received data packets. If the counter 27 for a priority class has the same count value as the ASN in a received data packet belonging to that priority class and a data packet in that priority class having a TSN equal to the ASN is not present in the buffer (e.g., is missing), the control logic 25 will treat the data packet having that priority class and a TSN equal to the ASN as having been output to the next, upper processing layer. Namely, the control logic 25 will increment the counter 27 associated with that priority class. This effectively flushes that data packet.

Accordingly, the present invention eliminates stall in the re-sequencing of received data packets. Because an ASN is generated for acknowledged and aborted data packets, stalls caused by acknowledgement error and data packet abortions are quickly eliminated. Also, the ASN is included in the data packet (i.e., signaled in-band), and does not impact system overhead.

As will be understood by those skilled in the art, Fig. 1 is a logical block diagram, and the elements illustrated therein are implemented by a programmed microprocessor and associated memory, hardware, firmware and/or a combination thereof. For example, the buffers 4 in one embodiment are registers, and in another embodiment are portions of a random access memory.

It will also be appreciated from the above disclosure that the present invention is applicable to any re-sequencing methodology, and is not limited to the re-sequencing methodology described.

The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such modifications are intended to be included within the scope of the following claims.